

AD-A148 576

COPE/ADS (CONTROL PROGRAM FOR ENGINEERING SYNTHESIS)  
(AUTOMATED DESIGN S. (U) NAVAL POSTGRADUATE SCHOOL  
MONTEREY CA G N VANDERPLATS JUN 84 NPS69-84-009

1/1

UNCLASSIFIED

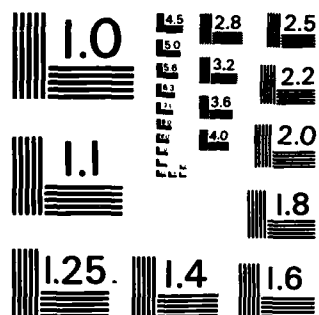
F/G 9/2

NL

END

FILED

DATE



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A148 576

NPS69-84-009

# NAVAL POSTGRADUATE SCHOOL

Monterey, California



DTIC  
ELECTE  
DEC 5 1984  
A

COPES/ADS -

A FORTRAN CONTROL PROGRAM FOR ENGINEERING SYNTHESIS

USING THE ADS OPTIMIZATION PROGRAM

by

Garret N. Vanderplaats

June 1984

Approved for public release; distribution unlimited.

Prepared for:

Naval Postgraduate School  
Monterey, California 93943

84 11 28 031

DTIC FILE COPY

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS69-84-009	2. GOVT ACCESSION NO. <b>AD-A148576</b>	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COPES/ADS - A FORTRAN CONTROL PROGRAM FOR ENGINEERING SYNTHESIS USING THE ADS OPTIMIZATION PROGRAM		5. TYPE OF REPORT & PERIOD COVERED
7. AUTHOR(s) Garret N. Vanderplaats		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE June 1984
		13. NUMBER OF PAGES 75
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  Optimization, Design, Synthesis		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  A FORTRAN Control Program for Engineering Synthesis (COPES) is presented for solving engineering design problems. This program uses the general-purpose optimization code ADS (Automated Design Synthesis) to provide a versatile design optimization capability. The program minimizes or maximizes a numerically defined objective function subject to a set of constraint limits. In addition to direct optimization, COPES includes approximation techniques as well as the capability to perform conventional parametric studies.		

DD FORM 1473  
1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE  
S/N 0102-LF-014-6/01

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Abstract (Cont'd.)...

Programming guidelines are presented, followed by sample input data and output for the various program options.

2/

S. N 0102- LF- 014- 6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

COPE/ADS - A FORTRAN CONTROL PROGRAM FOR ENGINEERING SYNTHESIS

USING THE ADS OPTIMIZATION PROGRAM

by

Garret N. Vanderplaats

Naval Postgraduate School

Monterey, California

June, 1984



Distribution For	
FORM 88A&I	<input checked="" type="checkbox"/>
TAB	<input checked="" type="checkbox"/>
Announced	<input type="checkbox"/>
Classification	
Distribution/	
Availability Codes	
at	Special

AI

## CONTENTS

	<u>PAGE</u>
ABSTRACT	111
1.0 INTRODUCTION	1
1.1 Design Example	2
2.0 PROGRAMMING GUIDELINES	7
3.0 COPEs DATA MANAGEMENT	9
4.0 COPEs TERMINOLOGY	10
5.0 COPEs DATA	11
5.1 Unformatted Data Input	11
5.2 Optimization Example	13
5.3 Copes Data Input	15
6.0 SAMPLE COMPUTER SOLUTIONS	35
7.0 REFERENCES	62
APPENDIX A: ADS PROGRAM OPTIONS	63
APPENDIX B: OVER-RIDING ADS DEFAULT PARAMETERS	70

## FIGURES

1. Three-Bar Truss	2
2. COPEs Example, Analysis only	36
3. COPEs Example, Optimization	38
4. COPEs Example, Optimization Using Approximation Techniques	42
5. COPEs Example, Parametric Study	58
6. COPEs Example, Combined Data for All Options	61

## TABLES

	<u>PAGE</u>
A.1 STRATEGY OPTIONS	63
A.2 OPTIMIZER OPTIONS	63
A.3 ONE-DIMENSIONAL SEARCH OPTIONS	64
A.4 PROGRAM OPTIONS	65
A.5 QUICK REFERENCE TO ADS OPTIONS	66
B.1 REAL PARAMETERS STORED IN ARRAY WK	70
B.2 DEFINITIONS OF REAL PARAMETERS CONTAINED IN ARRAY WK	71
B.3 INTEGER PARAMETERS STORED IN ARRAY IWK	72
B.4 DEFINITIONS OF INTEGER PARAMETERS CONTAINED IN ARRAY IWK	73



# ABSTRACT

A FORTRAN Control Program for Engineering Synthesis (COPES) is presented for solving engineering design problems. This program uses the general-purpose optimization code ADS (Automated Design Synthesis) to provide a versatile design optimization capability. The program minimizes or maximizes a numerically defined objective function subject to a set of constraint limits. In addition to direct optimization, COPES includes approximation techniques as well as the capability to perform conventional parametric studies. Programming guidelines are presented, followed by sample input data and output for the various program options.

## 1.0 INTRODUCTION

Most design processes require the minimization or maximization of some parameter which may be called the design objective. For the design to be acceptable, it must satisfy a variety of limitations which are referred to as constraints. While much of the practical design task cannot be automated, the engineering tasks which are quantifiable are usually well suited for computer solution.

To the extent that the problem can be stated in numerical terms, a computer program can be written to perform the necessary calculations. For this reason, computer analysis is commonplace in most engineering organizations. For example, in structural design, the materials, loads, and configuration may be defined, and a finite element analysis computer code is used to calculate stresses, deflections, frequencies and other response quantities of interest. If any of these parameters are not within the prescribed bounds, the engineer may change the structural member sizes or geometry, and re-run the program. The computer code therefore provides only the analysis of a proposed design, with the engineer making the actual design decisions. This approach to design, which may be called computer-aided design is commonly used today.

Another common use of analysis codes is in parametric studies. For example, an aircraft trajectory analysis code may be run repetitively for several payloads, calculating the aircraft range, to determine the range-payload dependency.

A logical extension to computer-aided design is fully automated design, where the computer makes the actual design decisions, or to perform parametric studies with minimum man-machine interaction. The purpose of the COPES program is to provide this automated design and parametric capability. The user must provide a FORTRAN program, in subroutine form, for analysis of the particular problem being solved. This analysis program is written according to a simple set of guidelines so that it can be easily coupled to the COPES program for automated design synthesis.

The program described here is a modification of the earlier version of the COPES program which used the CONMIN optimization program (refs. 1, 2). The principal differences are in the use of the ADS optimization program (refs. 3, 4), deletion of the two-variable function space option, optimum sensitivity, and minor changes in the data input.

This document describes the capabilities of the COPES program and its usage. A simple design example is first presented to demonstrate the program capabilities. Guidelines are given for writing analysis codes which can be coupled directly to COPES. Finally the data organization is outlined and sample data is presented. It is assumed that the user of COPES is familiar with the ADS user's manual (ref. 4), or at least has that manual available for ready reference.

The most commonly used option of COPES is for design optimization. Two approaches are available for this purpose. The first (NCALC=2) is direct optimization of the function by the ADS optimization subprogram. An alternative to this is through the use of approximation techniques (NCALC=6, ref. 5). This second option is often more

efficient for problems of under six design variables, but which require costly analysis, especially if multiple optimizations are performed. Also, this option is useful for design where the analysis information is not actually provided by a FORTRAN program, but is instead the result of experimentation (ref. 6).

In the following section, a simple example using the classical three-bar truss (ref. 7) is given to demonstrate problem formulation and solution using COPES/ADS.

### 1.1 Design Example

Assume it is required to design the simple planar truss shown in Figure 1. The objective is to find the minimum volume of material which will support the two (independent) loading conditions shown.

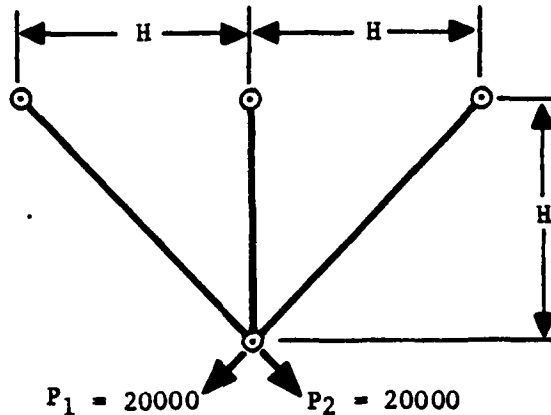


Figure 1: Three-Bar Truss

The design problem is stated as follows: Find the cross-sectional areas  $A_1$  and  $A_2$  (with  $A_3 = A_1$  for symmetry) which will

$$\text{Minimize} \quad \text{VOL} = H[\sqrt{2}A_1 + A_2 + \sqrt{2}A_3] \quad (1)$$

$$\text{Subject to;} \quad -15,000 \leq \sigma_{ij} \leq 20,000 \quad i=1,3 \quad j=1,2 \quad (2)$$

$$A_1 \geq 0.01 \quad i=1,3 \quad (3)$$

$$A_3 = A_1 \quad (4)$$

where  $\sigma_{ij}$  is the stress in member  $i$  under load condition  $j$ .

The equations necessary to calculate the member stresses are as follows:

Loading Condition 1

$$\sigma_{11} = \frac{P_1}{A_1} - \frac{A_2 A_3 P_1}{A_1 (A_1 A_2 + A_2 A_3 + 2 A_1 A_3)}$$

$$\sigma_{21} = \frac{2 A_3 P_1}{A_1 A_2 + A_2 A_3 + 2 A_1 A_3}$$

$$\sigma_{31} = \frac{-A_2 P_1}{A_1 A_2 + A_2 A_3 + 2 A_1 A_3}$$

Loading Condition 2

$$\sigma_{12} = \frac{-A_2 P_1}{A_1 A_2 + A_2 A_3 + 2 A_1 A_3}$$

$$\sigma_{22} = \frac{2 A_1 P_2}{A_1 A_2 + A_2 A_3 + 2 A_1 A_3}$$

$$\sigma_{32} = \frac{P_2}{A_3} - \frac{A_1 A_2 P_2}{A_3 (A_1 A_2 + A_2 A_3 + 2 A_1 A_3)}$$

Equation (1) is the objective function (volume) and Equations (2) are constraints on stress in the members. Equation (3) represents "side constraints" on the members to prevent the optimizer from proposing a design with negative member areas. Equation (4) is a design variable "linking" equation to maintain symmetry. The units on dimensions, stresses and loads are assumed to be consistent and the units used here are understood to be inches and pounds.

This design problem may be stated in standard mathematical form for optimization as:

$$\text{Minimize } F(X) \quad (5)$$

$$\text{Subject to; } G_j(X) \leq 0 \quad j=1,m \quad (6)$$

$$X_1^l \leq X_1 \leq X_1^u \quad i=1,n \quad (7)$$

where  $X$  is a vector containing the independent design variables,  $A_1$  and  $A_2$  and the  $G_j(X)$  are the constraints defined by Equations (2). The objective and constraints are functions of the design variables, but need not be explicit as given here. These functions can be evaluated by any means at our disposal including implicit iteration techniques or experimentation. It is assumed, however, that the functions are continuous and are defined everywhere in the range between  $X_1^l$  and  $X_1^u$ . The bounds on the design variables  $X$  given by Equation (7) are referred

to as side constraints and limit the region of search for the optimum. If any side constraint is not needed, its value should be set to a large negative or positive value for  $X_1^l$  or  $X_1^u$ , respectively. While the three-bar truss problem given above is straightforward, its solution is not trivial because it is not known which constraints will be critical [i.e. which  $G_j(\mathbf{X}) = 0$ ] at the optimum.

One design approach is to assume that each member will be stressed to its limit under at least one of the loading conditions. The stresses are calculated and each member is resized to its allowable stress. This must be done iteratively because the stresses are functions of the design variables and so change as the  $A_i$  are changed. The design obtained by this method is  $A_1 = A_3 = 1.0 \text{ in.}^2$  and  $A_2 = 0.0 \text{ in.}^2$ , with a volume of  $38.2 \text{ in.}^3$ . Note, however, that the objective function played no part in the design decisions and so there is no assurance that this is actually a minimum volume design. More importantly, it is desirable to devise techniques for optimum design of systems which may be defined by more than two variables and by much more complex analysis, where the optimum cannot easily be found by trial and error.

The COPES program provides this general capability by the use of the optimization program ADS. To use this design capability, a FORTRAN code must be provided which will calculate the various parameters. In writing the analysis code: 1) it is written in subroutine form with SUBROUTINE ANALIZ (ICALC) as the main routine, 2) it is segmented into INPUT, EXECUTION and OUTPUT and 3) all parameters which may be design variables, objective functions or constraints are contained in a single labeled common block called GLOBCM. The principal motivation for the COPES program is that the user does not have to identify the design variables, objective or constraints in advance and that the user does not have to formulate the problem in the standard form for optimization given by Equations (5-7). The design parameters are defined by data at the time of execution and the analysis information is converted by COPES to the standard form for use in ADS. Thus the COPES program greatly simplifies the use of optimization techniques for routine design tasks.

To demonstrate the simplicity with which a design-oriented analysis code can be written, the following FORTRAN subroutine was written for the analysis of the three-bar truss in Figure 1. Here  $P_1 = P_2 = P$ .

```

SUBROUTINE ANALIZ (ICALC)
COMMON /GLOBCM/ A1,A2,A3,VOL,SIG11,SIG21,SIG31,SIG12,SIG22,SIG32
IF (ICALC.GT.1) GO TO 10
C --- INPUT OR INITIALIZATION.
A1=1.
A2=1.
A3=1.
E=10.0E+6
H=10.
P=20000.
RETURN
C --- EXECUTION
10 CONTINUE
VOL=H*((A1+A3)*SQRT(2.)+A2)
DENOM=A1*A2+A2*A3+SQRT(2.)*A1*A3
SIG11=P/A1-A2*A3*P/(A1*DENOM)
SIG21=SQRT(2.)*A3*P/DENOM
SIG31=-A2*P/DENOM
SIG12=SIG31
SIG22=SQRT(2.)*A1*P/DENOM
SIG32=P/A3-A1*A2*P/(A3*DENOM)
IF (ICALC.LT.3) RETURN
C --- OUTPUT.
WRITE(6,20) P,H,E
20 FORMAT (1H1,4X,22HTHREE-BAR TRUSS OUTPUT//5X,3HH =,E12.5/5X,
1 3HP =,E12.5/5X,3HE =,E12.5)
WRITE(6,30) A1,A2,A3
30 FORMAT (/5X,4HA1 =,E12.5/5X,4HA2 =,E12.5/5X,4HA3 =,E12.5)
WRITE(6,40) SIG11,SIG21,SIG31,SIG12,SIG22,SIG32
40 FORMAT (/5X,7HSIG11 =,E12.5/5X,7HSIG21 =,E12.5/5X,7HSIG31 =,
1 E12.5/5X,7HSIG12 =,E12.5/5X,7HSIG22 =,E12.5/5X,7HSIG32 =,E12.5)
WRITE(6,50) VOL
50 FORMAT (/5X,8HVOLUME =,E12.5)
RETURN
END

```

This routine may be executed as an analysis program by the following main program:

```

C MAIN PROGRAM TO EXECUTE SUBROUTINE ANALIZ.
DO 10 I=1,3,2
10 CALL ANALIZ (I)
STOP
END

```

Moreover, ANALIZ can be coupled directly to the COPES program to perform this same function, or to perform optimization or trade-off studies.

This subroutine was coupled to COPES and an optimization was performed (NCALC = 2) to give the following design:

THREE-BAR TRUSS

P = 20000  
H = 10  
E = 1.0E+7

A1 = 0.7847  
A2 = 0.4201  
A3 = 0.7847

SIG11 = 19996  
SIG21 = 14506  
SIG31 = -5490  
SIG12 = -5490  
SIG22 = 14506  
SIG32 = 19996

VOLUME = 26.396

The parameters, P, H and E were input to the program and the remaining parameters were calculated. The independent design variables, A1 and A2 were changed during the optimization and symmetry was imposed so that A3 = A1. Note that A3 could have been treated as an independent variable as well but, assuming symmetry of the structure was desired, this "linking" of the design variables simplifies the optimization task.

This design was achieved with 21 calls to ANALIZ with ICALC = 2 (that is, the design process required 21 separate analyses). This design could surely have been found with fewer analyses were it performed by hand calculations. However, once having written the analysis subroutine, numerous other designs may be obtained for different materials, loading, or design stresses with only minimal effort. Furthermore, the design obtained here with a volume of 26.4 in.<sup>3</sup> is below the design of 38.2 in.<sup>3</sup> found by ratioing the members according to their stresses as may be done in traditional design methods.

This simple design example underscores the power of numerical optimization techniques and the ease with which they may be applied. The key to efficient use of the COPES program is the requirement that the ANALIZ code be written in a standard format so that it may be coupled to COPES without modification. The following section contains guidelines for writing a design-oriented analysis code.

## 2.0 PROGRAMMING GUIDELINES

In developing any computer code for engineering analysis, it is prudent to write the code in such a way that it is easily coupled to a general synthesis program such as COPEs. Therefore, a general programming practice is outlined here which in no way inhibits the use of the computer program in its traditional role as an analytic tool, but allows for simple adaptation to COPEs. This approach is considered good programming practice and provides considerable flexibility of design options. Only five basic rules must be followed:

1. Write the code in subroutine form with the primary routine called as; SUBROUTINE ANALIZ (ICALC). The name ANALIZ is compatible with the COPEs program and ICALC is a calculation control. Note that subroutine ANALIZ may call numerous other subroutines as required to perform the necessary calculations.
2. Segment the program into INPUT, EXECUTION and OUTPUT. The calculation control, ICALC, will determine the portion of the analysis code to be executed.

ICALC = 1: The program reads all data required to perform the analysis. Also, any initialization of constants which will be used repetitively during execution is done here. This initial input information is printed here for later reference and for program debugging.

ICALC = 2: The program performs the execution phase of the analysis task. No data reading or printing is done here, except on user-defined intermediate files. Data may be printed here during program debugging, in which case it should be controlled by a print control parameter which is read during input. In this way, this print may be turned off after the program is debugged, but may be used again during future program expansion and debugging. The reason that printing is not normally allowed during execution is that when optimization is being done, the code will be called many times with ICALC = 2, resulting in voluminous print.

ICALC = 3: The results of the analysis are printed. Also, the essential input parameters which may have been changed during optimization should be printed here for easy reference. When ICALC = 3, it is good practice to execute the analysis also because the analysis information may not be up to date. This is because the optimization program may not choose the last design considered as the optimum.

In Summary, When:

- ICALC = 1, Read input data.
- ICALC = 2, Execute the analysis.
- ICALC = 3, Execute the analysis and print the results.



3. Store all parameters which may be design variables, objective functions or constraints in a single labeled COMMON block called GLOBCM. The order in which they are stored is arbitrary. Also, it is not necessary to identify the design variables, objective function and constraints in advance. Any parameters which may be used for optimization or parametric studies may be stored in GLOBCM. A listing of the COPES program should be checked to see how many parameters may be stored in GLOBCM (this is the dimensioned size of the array called ARRAY). Initial distribution of COPES/ADS allows for 500 parameters. If more storage is needed, this array can be dimensioned accordingly in COPES and the parameter NARRAY at the beginning of the program changed to the new dimensioned size of ARRAY.
4. During execution or output, no parameters which are read during input should be updated. For example, if variable Q is initialized during input, the execution segment must not update Q such as  $Q=Q+3.2$ . Instead, a new variable,  $Y=Q+3.2$  should be defined.
5. Write all programs in standard language, avoiding machine dependent capabilities. While this guideline is not essential to the use of the analysis code with the COPES program, it makes the analysis code much more transportable between different computer systems, a capability which usually justifies a slight reduction in efficiency on a given machine.

Adherence to these guidelines not only leads to a more readable and machine independent computer code, but allows this code to be coupled to the COPES program without modification.

Having written the analysis code, it may be executed either with a simple main program or within the COPES program to perform analysis. To insure that guideline 4 is followed, the main test program given below is recommended. Note that this program calls ANALIZ twice with ICALC=3 (assuming execution is also performed when ICALC=3) to show that the same result is obtained repetitively.

```

C    MAIN PROGRAM TO CHECK SUBROUTINE ANALIZ.
C    READ, EXECUTE AND PRINT.
      DO 10 ICALC=1,3,2
10   CALL ANALIZ (ICALC)
C    EXECUTE AND PRINT AGAIN TO BE SURE THE RESULTS ARE THE SAME.
      DO 20 ICALC=2,3,2
20   CALL ANALIZ (ICALC)
      STOP
      END

```

Once the analysis code has been written, it can be coupled to the COPES program without modification. If it is desired to perform a simple analysis using COPES, only three data cards are required for the COPES program, namely the TITLE card, the control parameter, NCALC = 1, and an END card. If the optimization or parametric analysis capabilities of COPES are to be used, additional data must be read. This data will identify which parameters in the global common block, GLOBCM, are used.

### 3.0 COPES DATA MANAGEMENT

In order to perform design operations, the COPES program must access the data in common block GLOBCM. This is done by defining the location in GLOBCM where a specified parameter resides. For example, consider the common block for the three-bar truss:

```
COMMON /GLOBCM/ A1,A2,A3,VOL,SIG11,SIG21,SIG31,SIG12,SIG22,SIG32
```

The volume of material, VOL, is the fourth parameter in the common block; that is, it resides in location 4, referred to as the global location number. Similarly the stress in bar 1 under loading condition 2, SIG12, is in global location 8. Thus, the parameters are referred to by their respective location numbers in global common.

For convenience in preparing data for the COPES program, a simple "CATALOG" of parameters may be defined. For the three-bar truss, this catalog would be:

GLOBAL LOCATION	FORTTRAN NAME	DEFINITION
1	A1	Cross-sectional area of member 1.
2	A2	Cross-sectional area of member 2.
3	A3	Cross-sectional area of member 3.
4	VOL	Volume of material in the structure.
5	SIG11	Stress in member 1 under load condition 1.
6	SIG21	Stress in member 2 under load condition 1.
7	SIG31	Stress in member 3 under load condition 1.
8	SIG12	Stress in member 1 under load condition 2.
9	SIG22	Stress in member 2 under load condition 2.
10	SIG32	Stress in member 3 under load condition 2.

As another example, consider a global common block containing arrays:

```
COMMON /GLOBCM/ A,Y(10),Q,C(2,2),H
```

The variable catalog for this common block is:

GLOBAL LOCATION	FORTTRAN NAME	DEFINITION
1	A	Area
2	Y(10)	Vector of Y-coordinates
12	Q	.
13	C(2,2)	.
17	H	etc.

The dimensions are given with the FORTRAN name as a reminder that the parameter is an array. In this case, the third parameter in the Y-array is in global location 4. Remembering that arrays are stored

column by column, the C(1,2) array location is in global location 15.

It will be seen that identifying parameters according to their location in GLOBCM provides a great deal of flexibility in using the COPES program for design.

#### 4.0 COPES TERMINOLOGY

The COPES program provides four specific capabilities:

1. Simple analysis, just as if COPES was not used.
2. Optimization - Minimization or maximization of one calculated parameter with limits imposed on other parameters.
3. Approximation techniques - Optimization using approximations. Often more efficient than standard optimization for up to six design variables or if multiple optimizations are to be performed.
4. Parametric analysis - The effect of changing one or more design variables on one or more calculated parameters.

In defining the data required to execute the COPES program, the following definitions are useful:

Design Variables - Those parameters which the optimization program is allowed to change in order to improve the design. Design variables appear only on the right hand side of equations in the analysis program. COPES considers two types of design variables, independent and dependent. If two or more variables are always required to have the same value or be in a constant ratio, one is the independent variable while the remaining are dependent variables. For example, in the three-bar truss, A1 may be the independent variable, while A3=A1 is a dependent variable, treated this way to insure symmetry in the design.

Objective Function - The parameter which is to be minimized or maximized during optimization. Objective functions always appear on the left side of equations, unless the objective function is also a design variable (i.e. the cross-sectional area of member A1 of the three bar truss may be minimized as the objective if there is a particular reason to do so. In that case, A1 would be both a design variable and the objective function). An objective function may be linear or nonlinear, implicit or explicit, but must be a function of the design variables to be meaningful.

Constraint - Any parameter which must not exceed specified bounds for the design to be acceptable. Constraint functions always appear on the left side of equations. Just as for objective functions, constraints may be linear or nonlinear, implicit or explicit, but must be functions of the design variables.

Constraint Set - A group of constraints which appear consecutively in the global common block, GLOBCM, and which all have the same limits imposed. This is a convenience which allows several constraints to be identified with a minimum of data.

Global Common - Common block GLOBCM containing design information.

Global Location - Location of a particular parameter in GLOBCM.

Strategy - Optimization strategy such as Sequential Quadratic Programming .

Optimizer - Actual algorithm to perform the optimization.

One-Dimensional Search - Line search routine used by the Optimizer.

## 5.0 COPES DATA

The COPES program reads data from unit 5 and writes output to unit 6. Units 10 and 11 are used as scratch files. The scratch file numbers may be changed by changing two cards at the beginning of the COPES program. The FORTRAN names defining the file numbers are ISCR1 and ISCR2.

In order to execute the COPES program, it is necessary to provide data for COPES, followed by data for the ANALIZ program which is coupled to COPES. This section defines the data which is required by COPES. The data is segmented into "BLOCKS" for convenience. All formats are alpha-numeric for TITLE and END cards, F10 for real data and I10 for integer data. Unformatted input is allowed for COPES data.

The COPES data begins with a TITLE card and ends with an END card. This is followed by data to be read by the user supplied subroutine ANALIZ when ICALC = 1.

Comment cards may be inserted anywhere in the COPES data stack prior to the END card, and are identified by a dollar sign (\$) in column 1 of the card.

### 5.1 Unformatted Data Input

While the user's sheet defines COPES data in formatted fields of ten, the data may actually be read in a simplified fashion by separating data by commas or by one or more blanks. If more than one number is contained on an unformatted data card, a comma must appear somewhere on the card. If exponential numbers such as 2.5+10 or 2.5E+10 (depending on the computer system) are read on an unformatted card, there must be no embedded blanks. Unformatted cards may be intermingled with formatted cards. Real numbers on an unformatted card should have a decimal point.

#### Examples

Unformatted data:

5,7,1.3,1.0+20,0,-5.1

5,7,1.3,1.0+20,, -5.1

5 7 1.3 1.0+20,, -5.1

5        7    1.3, 1.0+20 0 -5.1

Equivalent formatted data:

Col	10	20	30	40	50	60	70	80
	5	7	1.3	1.0+20	0	-5.1		

Unformatted data:

2

2,3

2 3

Equivalent formatted data:

Col	10	20	30
	2		
	2	3	

Note. This data has been right justified.

2 3

Note. This data contains no commas, so it is assumed to be formatted already.

Unformatted data:

1,2,3,4,5,6,7,8,9,10,11

Equivalent formatted data:

Col	10	20	30	40	50	60	70	80
	1	2	3	4	5	6	7	8
	9	10	11					

Note that two formatted cards are created here.

Unformatted data:

1,2,3,4,5,6

7,8,9,10,11

Equivalent formatted data:

Col	10	20	30	40	50	60	70	80
	1	2	3	4	5	6		
	7	8	9	10	11			

Note that the above two examples do not produce the same formatted data cards. If more than eight entries are given on an unformatted card, they will spill to the next card, but each unformatted card begins a new equivalent formatted card.

## 5.2 Optimization Example

For the three-bar truss example, it is now a simple matter to create the COPES data to solve the design problem. An essential part of this is problem formulation. Therefore, the basic design questions are first presented, followed by the necessary data to solve this problem using COPES/ADS.

### What Must We Know?

Number of independent design variables:

A1, A2 (A3=A1) NDV=2

↑  
└ Dependent Variable

Total number of design variables:

A1, A2, A3 NDVTOT=3

What are the locations of the design variables?

1, 2 and 3

What is the objective?

Minimize VOL

↑  
└ Location 4

SGNOPT=-1.0 (+1.0 would maximize)

Constraints?

Yes,  $-15,000 \leq \text{SIGLJ} \leq 20,000$

1 Set, Locations 5-10, Nonlinear inequality

### COPEs Data to Perform Optimization

The data is simply listed here. Further definitions of the parameters are given in the data block input in the following section.

\$ BLOCK A: TITLE

3-BAR TRUSS DESIGN EXAMPLE

\$ BLOCK B:

2,2 Optimization, 2 independent variables

\$ BLOCK C:

0,4,8,2020 Strategy, Optimizer, One-Dimensional Search,  
Print control

\$ BLOCK D: OMIT

\$ BLOCK E: OMIT

\$ BLOCK F:

3,4,-1.0 NDVTOT, IOBJ, SGNOPT

\$ BLOCK G:

0.1,1.0E+20 Design Variable Bounds  
0.1,1.0E+20

\$ BLOCK H:

1,1 Design Variable Number and Global Location  
2,2  
1,3

\$ BLOCK I:

1 Number of Constraint Sets

\$ BLOCK J:

5,10 Global Locations of Constrained Variables  
-15000.,0.,20000. Lower and Upper Bounds

\$ BLOCK R:

END Required END card

This data was used, together with the ANALIZ subroutine in Section 1.1 to create the design given there. This example is used in Section 6.0 to demonstrate all basic options of COPEs and sample output is given there.

### 5.3 Copes Data Input

The necessary input data to the COPEs program is given here. Note that, depending on the program options to be used, much of this data may be omitted.

#### DATA BLOCK A

DESCRIPTION: Title Card.

#### FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
TITLE								20A4
THREE-BAR TRUSS								

#### FIELD

#### CONTENTS

1-8 Any title may be given on this card, up to 80 characters.



## DATA BLOCK B

DESCRIPTION: Program Control Parameters.

### FORMAT AND EXAMPLE

1	2	3	4	5	6	FORMAT
NCALC	NDV	NAPRX	NPV	IPNPUT	IPDBG	6I10
2	2	1	3	0	0	

### FIELD

### CONTENTS

- 1 NCALC: Calculation Control
  - 0 - Read input and stop. Data of blocks A, B and R is required. Remaining data is optional.
  - 1 - One cycle through the program. The same as executing ANALIZ as a stand-alone program. Data of blocks A, B and R is required. Remaining data is optional.
  - 2 - Optimization. Data of blocks A-J and R is required. Remaining data is optional.
  - 3 - Optimization Using Approximation Techniques. Data of blocks A-O and V is required. Remaining data is optional.
  - 4 - Parametric Study. Data of blocks A, B and P-V is required. Remaining data is optional.
- 2 NDV: Number of independent design variables for optimization.
- 3 NAPRX: Parameter to indicate if data is to be read for optimization using approximation techniques.
  - 0 - No data will be read.
  - 1 - Data will be read.
- 4 NPV: Number of parameters to be varied in a parametric study.
- 5 IPNPUT: Input data print control.
  - 0 - Print card images of data plus formatted print of input information.
  - 1 - Formatted print of input information only.
  - 2 - No print of input data.
- 6 IPDBG: Debug print control.

### REMARKS

1) Data to be read in the following blocks depends on the values of NDV, NAPRX and NPV. Actual program execution is controlled by the value of NCALC, and it is assumed by COPES that the necessary data has been provided.

DATA BLOCK C: OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Integer optimization control parameters.

FORMAT AND EXAMPLE

1	2	3	4	5	6		FORMAT
ISTRAT	IOPT	IONED	IPRINT	IOVRDR	IOVRDI		6I10
0	5	7	2020	0	0		

FIELD

CONTENTS

- 1 Strategy option to be used. DEFAULT = 0.
- 2 Optimizer option to be used. DEFAULT = 5.
- 3 One-Dimensional search option to be used. DEFAULT = 7.
- 4 ADS Print control.
- 5 Number of real ADS parameters to be over-ridden. Usually IOVRDR = 0 is adequate so the ADS default parameters will be used.
- 6 Number of integer ADS parameters to be over-ridden. Usually IOVRDI = 0 is adequate so the ADS default parameters will be used.

REMARKS

- 1) See Appendix A for available options for ISTRAT, IOPT and IONED, as well as guidelines for choosing these parameters.
- 2) See Appendix B for ADS internal parameters which may be modified by the user to "fine tune" the optimization to a specific class of problems.
- 3) See the ADS User's Manual (ref. 4) for further details.

DATA BLOCK D: OMIT IF IOVRDR = 0 IN BLOCK C

DESCRIPTION: Identifiers and values of real parameters to be input.

FORMAT AND EXAMPLE

1	2	FORMAT
ILOC	VALUE	I10,F10
11	0.002	

FIELD

CONTENTS

- 1 ILOC: Location in array WK where this parameter is stored.
- 2 VALUE: New value of the parameter.

REMARKS

- 1) See Appendix B for parameter default values, definitions and locations.
- 2) See the ADS User's Manual (ref. 4) for further details.

DATA BLOCK E: OMIT IF IOVRDI = 0 IN BLOCK C

DESCRIPTION: Identifiers and values of integer parameters to be input.

FORMAT AND EXAMPLE

1            2		FORMAT
ILOC	IVALUE	2I10
2	1	

FIELD

CONTENTS

- 1    ILOC:    Location in array IWK where parameter is stored.
- 2    IVALUE: New value of the parameter.

REMARKS

- 1) See Appendix B for default values, definitions and locations.
- 2) See the ADS User's Manual (ref. 4) for further details.

DATA BLOCK F: OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Total number of design variables, design objective global location and sign.

FORMAT AND EXAMPLE

1	2	3	FORMAT
NDVTOT	IOBJ	SGNOPT	2I10,F10
3	4	-1.0	

FIELD

CONTENTS

- 1 NDVTOT: Total number of design variables, including both dependent and independent variables. This option allows two or more parameters to be assigned to a single design variable. The value of each parameter is the value of the design variable times a multiplier, which may be different for each parameter. DEFAULT = NDV.
- 2 IOBJ: Global location of the objective function.
- 3 SGNOPT: Sign used to identify whether the objective function is to be maximized or minimized. +1.0 indicates maximization. -1.0 indicates minimization. If SGNOPT is not unity in magnitude, it acts as a multiplier as well, to scale the magnitude of the objective.

REMARKS

- 1) If maximization is to be performed, the objective will be changed internally and the objective printed by ADS will appear opposite in sign. This is because ADS minimizes the objective, so if maximization is performed, ADS will minimize the negative of the true objective.

DATA BLOCK G: OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Design variable bounds and initial values.

FORMAT AND EXAMPLE

1	2	3		FORMAT
VLBI	VUBI	XI		3F10
0.01	1.0+20	0.0		

NOTE: READ ONE CARD FOR EACH OF THE NDV INDEPENDENT DESIGN VARIABLES.

FIELD

CONTENTS

- 1 VLBI: Lower bound on the design variable. If no lower bound is to be imposed, set VLBI to a large negative value.
- 2 VUBI: Upper bound on the design variable. If no upper bound is to be imposed, set VUBI to a large positive value.
- 3 XI: Initial value of the design variable. Is XI is non-zero, this will supercede the value initialized by the user-supplied subroutine ANALIZ.

DATA BLOCK H: OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Design variable identification.

FORMAT AND EXAMPLE

1	2	3	FORMAT
NDSGN	IDSGN	AMULT	
1	1	1.0	2I10,F10

NOTE: READ ONE CARD FOR EACH OF THE NDVTOT DESIGN VARIABLES.

FIELD

CONTENTS

- 1 NDSGN: Design variable number associated with this parameter.
- 2 IDSGN: Global location number associated with this parameter.
- 3 AMULT: Constant multiplier on this parameter. The value of the parameter will be the value of the design parameter, NDSGN, times AMULT. DEFAULT = 1.0

REMARKS

- 1) The first parameter assigned with a given design variable is the independent variable. Following this, any parameters assigned to this design variable are dependent variables.
- 2) The value of AMULT associated with an independent variable should always be 1.0.

DATA BLOCK I: OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Number of constraint sets.

FORMAT AND EXAMPLE

1		FORMAT
NCONS		I10
1		

NOTE: NCONS MAY BE ZERO, INDICATING THE PROBLEM IS UNCONSTRAINED.

FIELD

CONTENTS

1 NCONS: Number of constraint sets in the optimization problem.

REMARKS

- 1) If two or more adjacent parameters in the global common block have the same limits imposed, these are part of the same constraint set.



DATA BLOCK J: OMIT IF NDV = 0 IN BLOCK B OR NCONS = 0 IN BLOCK I

DESCRIPTION: Constraint identification, bounds and normalization factors.

FORMAT AND EXAMPLE

1	2	3	4	FORMAT
ICON	JCON	LCON		3I10
4	8	0		
BL	FACT1	BU	FACT2	4F10
-15000.	1.0	20000.	1.0	

NOTE: READ TWO CARDS FOR EACH OF THE NCONS CONSTRAINT SETS.

FIELD

CONTENTS

- 1 ICON: First global location number corresponding to this constraint set.
- 2 JCON: Last global location number corresponding to this constraint set. DEFAULT = ICON.
- 3 LCON: Constraint type identifier.
  - 0,1 - Nonlinear inequality constraint.
  - 2 - Linear inequality constraint.
  - 1 - Nonlinear equality constraint.
  - 2 - Linear equality constraint.

FIELD

CONTENTS

- 1 BL: Lower bound on the constrained variables.  
If BL.LT.-1.0E+15, the lower bound is ignored by COPES.
- 2 FACT1: Normalization factor on the lower bound.  
DEFAULT = MAX of ABS(BL) and 1.0.
- 3 BU: Upper bound on the constrained variables.  
If BU.GT.1.0E+15, the upper bound is ignored by COPES.
- 4 FACT2: Normalization factor on the upper bound.  
DEFAULT = MAX of ABS(BU) and 1.0.

#### REMARKS

- 1) The normalization factor should usually be defaulted.
- 2) The constraint functions sent to ADS are of the form:  
 $(BL - VALUE)/FACT1 \leq 0$  and  $(VALUE - BU)/FACT2 \leq 0$ .
- 3) Each constrained parameter is converted to two constraints for ADS unless  $ABS(BL)$  or  $ABS(BU)$  exceeds  $1.0E+15$ , in which case no constraint is created for that bound.
- 4) In ADS Version 1.00, equality constraints are only allowed for  $ISTRAT = 1-5$ .

DATA BLOCK K: OMIT IF NAPRX = 0 IN BLOCK B

DESCRIPTION: Approximate analysis/optimization control parameters.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
IPAPRX	NXI	NXFI	NXA	INOM	ISCRX	ISCRXF		7I10
1	4	0	1	0	0	0		
KMIN	KMAX	NPMAX	JNOM	MAXTRM				5I10
0	0	0	0	0				

FIELD

CONTENTS

- 1 IPAPRX: Print control. Values of 0-4 with increasing amounts of print for larger IPAPRX.
- 2 NXI: Number of candidate X-vectors read as data.
- 3 NXFI: Number of X-F pairs read as data.
- 4 NXA: 0 or 1. If NXA = 1, the design variables read by Subroutine ANALIZ form a candidate X-vector.
- 5 INOM: Nominal X-vector. DEFAULT = Best available.
- 6 ISCRX: Unit from which the NXI X-vectors are read. DEFAULT = 5. If ISCRX is not 5 (the default), the data is assumed to be in binary format.
- 7 ISCRXF: Unit from which the NXFI X-F pairs are read. DEFAULT = 5. If ISCRXF is not 5 (the default), the data is assumed to be in binary format.

FIELD

CONTENTS

- 1 KMIN: Minimum number of approximation iterations. DEFAULT =  $2 \cdot \text{NDV} + 2 - \text{NXI} - \text{NXFI} - \text{NXA}$ .
- 2 KMAX: Maximum number of approximation iterations. DEFAULT =  $\text{NDV} + \text{NDV}(\text{NDV}+1)/2 + 5$ . If KMAX = -1, only one approximate optimization is performed and the resulting proposed optimum is printed, together with the approximating functions. This is useful when the precise functions are evaluated experimentally or by another program, and read as input to COPES for the next approximate optimization.

- 3     NPMAX: Maximum number of designs retained for approximations.  
          DEFAULT = ALL.
- 4     JNOM: Number of designs after which the best design is picked  
          as nominal. Before this the most recent design is  
          nominal. DEFAULT =  $2 \cdot \text{NDV} + 2$
- 5     MAXTRM: Terms retained in the Taylor series approximation.  
          1 - Linear terms only.  
          2 - Linear terms plus diagonal elements of the Hessian  
          matrix.  
          3 - Full second-order expansion.  
          DEFAULT = 3.

DATA BLOCK L: OMIT IF NAPRX = 0 IN BLOCK B

DESCRIPTION: Bounds for approximate optimization.

FORMAT AND EXAMPLE

	2	3	4	5	6	7	8	FORMAT
DX1	DX2	DX3	DX4	DX5	...	...	...	8F10
0.2	0.2							

NOTE: TWO OR MORE CARDS MAY BE READ HERE FOR A TOTAL OF NDV ENTRIES.

FIELD

CONTENTS

1-8 DXI: Allowable change (in magnitude) of the Ith design variable during each approximate optimization.

DATA BLOCK M: OMIT IF NAPRX = 0 IN BLOCK B

DESCRIPTION: Multipliers on bounds for approximate optimization.

FORMAT AND EXAMPLE

1	2		FORMAT
XFACT1	XFACT2		2F10
0.0	0.0		

FIELD

CONTENTS

- 1 XFACT1: Multiplier on DXI when the diagonal elements of the Hessian matrix are available. DEFAULT = 1.5.
- 2 XFACT2: Multiplier on DXI when all elements of the Hessian matrix are available. DEFAULT = 2.0.

REMARKS

- 1) These factors increase the move limits given in Block L, based on the assumption that larger move limits are possible when nonlinear information is available.

DATA BLOCK N: OMIT IF NAPRX = 0 IN BLOCK B OR NXI = 0 IN BLOCK K

DESCRIPTION: Candidate X-Vectors for approximate optimization.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
XI1	XI2	XI3	XI4	...	...	...	...	8F10
0.8	0.6							

NOTE: NXI SETS OF DATA ARE READ HERE.

NOTE: MORE THAN ONE CARD MAY BE READ FOR EACH SET OF DATA.

FIELD

CONTENTS

1-8 XIJ: Jth value if the Ith X-vector, J=1,NDV.

REMARKS

- 1) If ISCRX is not 5 (the default value), data read in this block is assumed to be in binary format.

DATA BLOCK 0: OMIT IF NAPRX = 0 IN BLOCK B OR NXFI = 0 IN BLOCK K

DESCRIPTION: X-F pairs of information for approximate optimization.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
X1	X2	X3	X4	...	...	...	...	8F10
0.8	0.6							
F1	F2	F3	F4	F5	...	...	...	8F10
26.529	20000.	13474.	-6525.7	-6525.7				

NOTE: NXFI SETS OF DATA ARE READ HERE.

NOTE: MORE THAN ONE CARD MAY BE REQUIRED FOR XI OR FI.

NOTE: NDV VALUES OF X AND NF VALUES OF F ARE READ FOR EACH SET OF DATA,  
WHERE NF = 1 + THE TOTAL NUMBER OF CONSTRAINED PARAMETERS  
(NOT CONSTRAINT SETS).

FIELD

CONTENTS

- 1 XI: Ith value of the X-vector, I=1,NDV.
- 2 FI: Ith function value, I=1,NF.

REMARKS

- 1) If ISCRXF is not 5 (the default value), data read in this block is assumed to be in binary format.



DATA BLOCK P: OMIT IF NPV = 0 IN BLOCK B

DESCRIPTION: Parametric variables.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NPF	IPRNT							2I10
3	0							
NPN1	NPN2	NPN3	NPN4	...	...	...	...	8I10
1	2							

NOTE: TWO OR MORE CARDS ARE READ HERE.

FIELD

CONTENTS

- 1 NPF: Number of separate functions to be calculated in the parametric study.
- 2 IPRNT: Debug print control. If greater than zero, detailed print will be called at each step in the parametric study. DEFAULT = no print.

FIELD

CONTENTS

- 1-8 NPNI: Global location associated with the I-th function to be evaluated in the parametric study.

REMARKS

- 1) More than eight functions are allowed. Add data cards as necessary to contain data.

DATA BLOCK Q: OMIT IF NPV = 0 IN BLOCK B

DESCRIPTION: Variables to be changed in the parametric study.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
IPV	NIPV							2I10
3	0							
VP1	VP2	VP3	VP4	...	...	...	...	8I10
1	2							

NOTE: READ ONE SET OF DATA FOR EACH OF THE NPV PARAMETRIC VARIABLES.

NOTE: TWO OR MORE DATA CARDS ARE READ FOR EACH SET OF DATA.

FIELD

CONTENTS

- 1 IPV: Global variable number associated with this parametric variable.
- 2 NIPV: Number of separate values of this variable to be read on the next card.

FIELD

CONTENTS

- 1-8 VPI: I-th value of the parameter. I = 1 is the nominal value about which the parametric study is performed.

REMARKS

- 1) More than eight values of the parameter are allowed. Add data cards as required to contain the data.

DATA BLOCK R

DESCRIPTION: COPEs data "END" card.

FORMAT AND EXAMPLE

1		FORMAT
END		3A1
END		

FIELD

CONTENTS

1      END:    The word "END" in columns 1-3.

REMARKS

- 1) This card MUST appear at the end of the COPEs data.
- 2) This ends the COPEs input data.
- 3) Data for the user-supplied subroutine ANALIZ follows this.

## 6.0 SAMPLE COMPUTER SOLUTIONS

Sample solutions are presented here for the various COPES program options defined by the parameter NCALC in DATA BLOCK B. The ANALIZ subroutine given in Section 1.1 was used to produce the results given here. Note that Subroutine ANALIZ does not read data, so only COPES data is required here. In the usual case where data is read as input to the analysis routine, this data would follow directly after the COPES "END" card.

The output from a COPES program execution includes a title page followed by a copy of the input data. Then the required program executions are performed and the final output information is printed. Figures 2-5 contain output for all options of COPES as follows:

<u>NCALC</u>	<u>FIGURE</u>	<u>PROBLEM SOLVED</u>
1	2	Analysis only.
2	3	Optimization by ADS.
3	4	Optimization using approximation techniques.
4	5	Parametric study.

Note that data for the COPES program is read depending on the value of the parameters NDV, NAPRX and NPV in DATA BLOCK B. The actual program execution is determined by the value of NCALC. Therefore, data may be read for all program options even though only one option of the program is to be executed. Figure 6 is a copy of the combined data for all the previous examples. This data may be used for any program option simply by changing the value of NCALC.

CCCCCCC	0000000	PPPPPPP	EEEEEEE	SSSSSSS
C	0 0	P P	E	S
C	0 0	P P	E	S
C	0 0	PPPPPP	EEEE	SSSSSSS
C	0 0	P	E	S
C	0 0	P	E	S
CCCCCCC	0000000	P	EEEEEEE	SSSSSSS

C O P E S / A D S  
C O N T R O L P R O G R A M  
F O R  
E N G I N E E R I N G S Y N T H E S I S  
U S I N G T H E  
A D S O P T I M I Z A T I O N P R O G R A M

T I T L E

THREE-BAR TRUSS

CARD IMAGES OF INPUT DATA

CARD IMAGE

- 1) THREE-BAR TRUSS
- 2) 1,
- 3) END

Figure 2: COPEs Example, Analysis Only

TITLE:  
THREE-BAR TRUSS

CONTROL PARAMETERS;  
CALCULATION CONTROL, NCALC = 1  
NUMBER OF GLOBAL DESIGN VARIABLES, NDV = 0  
INPUT PARAMETER FOR APPROX. OPT., NAPRX = 0  
NUMBER OF PARAMETRIC VARIABLES, NPV = 0  
INPUT INFORMATION PRINT CODE, IPNPUT = 0  
DEBUG PRINT CODE, IPDBG = 0

CALCULATION CONTROL, NCALC  
VALUE MEANING  
1 SINGLE ANALYSIS  
2 OPTIMIZATION  
3 APPROXIMATE OPTIMIZATION  
4 PARAMETRIC STUDY

\* \* ESTIMATED DATA STORAGE REQUIREMENTS

REAL			INTEGER		
INPUT	EXECUTION	AVAILABLE	INPUT	EXECUTION	AVAILABLE
7	7	5000	1	1	1000

THREE-BAR TRUSS OUTPUT

A1 = 0.10000E+01  
A2 = 0.10000E+01  
A3 = 0.10000E+01

STRESSES  
SIG11 = 0.14142E+05  
SIG21 = 0.82843E+04  
SIG31 = -0.58579E+04  
SIG12 = -0.58579E+04  
SIG22 = 0.82843E+04  
SIG32 = 0.14142E+05

VOLUME = 0.38284E+02  
PROGRAM CALLS TO ANALIZ

ICALC	CALLS
1	1
2	1
3	1

Figure 2 Concluded: COPES Example, Analysis Only

CCCCCCC	0000000	PPPPPPP	EEEEEEE	SSSSSSS
C	0 0	P P	E	S
C	0 0	P P	E	S
C	0 0	PPPPPPP	EEEE	SSSSSSS
C	0 0	P	E	S
C	0 0	P	E	S
CCCCCCC	0000000	P	EEEEEEE	SSSSSSS

C O P E S / A D S  
C O N T R O L P R O G R A M  
F O R  
E N G I N E E R I N G S Y N T H E S I S  
U S I N G T H E  
A D S O P T I M I Z A T I O N P R O G R A M

T I T L E

THREE-BAR TRUSS

CARD IMAGES OF INPUT DATA

CARD	IMAGE
1)	THREE-BAR TRUSS
2)	\$ BLOCK B
3)	2,2
4)	\$ BLOCK C
5)	0,5,7,0
6)	\$ BLOCK D
7)	3,4,-1.
8)	\$ BLOCKS E AND F NOT NEEDED
9)	\$ BLOCK G
10)	.01,1.0+20
11)	.01,1.0+20
12)	\$ BLOCK H
13)	1,1
14)	2,2
15)	1,3
16)	\$ BLOCK I
17)	1
18)	\$ BLOCK J
19)	5,10
20)	-15000.,,20000.
21)	\$ BLOCKS K-Q NOT NEEDED
22)	\$ BLOCK R
23)	END

Figure 3: COPES Example, Optimization

TITLE:  
THREE-BAR TRUSS

CONTROL PARAMETERS;  
CALCULATION CONTROL, NCALC = 2  
NUMBER OF GLOBAL DESIGN VARIABLES, NDV = 2  
INPUT PARAMETER FOR APPROX. OPT., NAPRX = 0  
NUMBER OF PARAMETRIC VARIABLES, NPV = 0  
INPUT INFORMATION PRINT CODE, IPNPUT = 0  
DEBUG PRINT CODE, IPDBG = 0

CALCULATION CONTROL, NCALC  
VALUE MEANING  
1 SINGLE ANALYSIS  
2 OPTIMIZATION  
3 APPROXIMATE OPTIMIZATION  
4 PARAMETRIC STUDY

\* \* OPTIMIZATION INFORMATION

OPTIMIZATION CONTROL PARAMETERS

ISTRAT	IOPT	IONED	IPRINT	NSCAL	IOVRDR	IOVRDI
0	5	7	0	0	0	0

GLOBAL VARIABLE NUMBER OF OBJECTIVE = 4  
MULTIPLIER (NEGATIVE INDICATES MINIMIZATION) = -0.1000E+01

DESIGN VARIABLE INFORMATION  
NON-ZERO INITIAL VALUE WILL OVER-RIDE MODULE INPUT

D. V. NO.	LOWER BOUND	UPPER BOUND	INITIAL VALUE
1	0.10000E-01	0.11000E+16	0.0
2	0.10000E-01	0.11000E+16	0.0

DESIGN VARIABLES

ID	D. V. NO.	GLOBAL VAR. NO.	MULTIPLYING FACTOR
1	1	1	0.10000E+01
2	2	2	0.10000E+01
3	1	3	0.10000E+01

CONSTRAINT INFORMATION

THERE ARE 1 CONSTRAINT SETS

ID	GLOBAL VAR. 1	GLOBAL VAR. 2	TYPE ID	LOWER BOUND	NORM. FACTOR	UPPER BOUND	NORM. FACTOR
1	5	10	0	-0.1500E+05	0.1500E+05	0.2000E+05	0.2000E+05

TOTAL NUMBER OF CONSTRAINED PARAMETERS = 6

Figure 3 Continued: COPES Example, Optimization



\* \* ESTIMATED DATA STORAGE REQUIREMENTS

INPUT	REAL EXECUTION	AVAILABLE	INPUT	INTEGER EXECUTION	AVAILABLE
42	192	5000	25	59	1000

THREE-BAR TRUSS OUTPUT

A1 = 0.10000E+01  
A2 = 0.10000E+01  
A3 = 0.10000E+01

STRESSES

SIG11 = 0.14142E+05  
SIG21 = 0.82843E+04  
SIG31 = -0.58579E+04  
SIG12 = -0.58579E+04  
SIG22 = 0.82843E+04  
SIG32 = 0.14142E+05

VOLUME = 0.38284E+02

OPTIMIZATION RESULTS

OBJECTIVE FUNCTION

GLOBAL LOCATION 4 FUNCTION VALUE 0.26396E+02

DESIGN VARIABLES

ID	D. V. NO.	GLOBAL VAR. NO.	LOWER BOUND	VALUE	UPPER BOUND
1	1	1	0.10000E-01	0.78474E+00	0.11000E+1
2	2	2	0.10000E-01	0.42005E+00	0.11000E+1
3	1	3	0.10000E-01	0.78474E+00	0.11000E+1

DESIGN CONSTRAINTS

ID	GLOBAL VAR. NO.	LOWER BOUND	VALUE	UPPER BOUND
1	5	-0.15000E+05	0.19996E+05	0.20000E+05
3	6	-0.15000E+05	0.14506E+05	0.20000E+05
5	7	-0.15000E+05	-0.54903E+04	0.20000E+05
7	8	-0.15000E+05	-0.54903E+04	0.20000E+05
9	9	-0.15000E+05	0.14506E+05	0.20000E+05
11	10	-0.15000E+05	0.19996E+05	0.20000E+05

Figure 3 Continued: COPES Example, Optimization

### THREE-BAR TRUSS OUTPUT

A1 = 0.78474E+00  
A2 = 0.42005E+00  
A3 = 0.78474E+00

### STRESSES

SIG11 = 0.19996E+05  
SIG21 = 0.14506E+05  
SIG31 = -0.54903E+04  
SIG12 = -0.54903E+04  
SIG22 = 0.14506E+05  
SIG32 = 0.19996E+05

VOLUME = 0.26396E+02

### PROGRAM CALLS TO ANALIZ

ICALC	CALLS
1	1
2	21
3	2

Figure 3 Concluded: COPES Example, Optimization

CCCCCCC	0000000	PPPPPPP	EEEEEEE	SSSSSSS
C	0 0	P P	E	S
C	0 0	P P	E	S
C	0 0	PPPPPPP	EEEE	SSSSSSS
C	0 0	P	E	S
C	0 0	P	E	S
CCCCCCC	0000000	P	EEEEEEE	SSSSSSS

C O P E S / A D S  
 C O N T R O L P R O G R A M  
 F O R  
 E N G I N E E R I N G S Y N T H E S I S  
 U S I N G T H E  
 A D S O P T I M I Z A T I O N P R O G R A M

T I T L E

THREE-BAR TRUSS

Figure 4: COPES Example, Optimization Using Approximation Techniques

# CARD IMAGES OF INPUT DATA

CARD	IMAGE
1)	THREE-BAR TRUSS
2)	\$ BLOCK B
3)	3,2,1
4)	\$ BLOCK C
5)	0,5,7,0
6)	\$ BLOCK D
7)	3,4,-1.
8)	\$ BLOCKS E AND F NOT NEEDED
9)	\$ BLOCK G
10)	.01,1.0+20
11)	.01,1.0+20
12)	\$ BLOCK H
13)	1,1
14)	2,2
15)	1,3
16)	\$ BLOCK I
17)	1
18)	\$ BLOCK J
19)	5,10
20)	-15000.,,20000.
21)	\$ BLOCK K
22)	4,4
23)	0
24)	\$ BLOCK L
25)	.2,.2
26)	\$ BLOCK M
27)	0.
28)	\$ BLOCK N
29)	.8,.6
30)	.4,1.
31)	1.,.2
32)	.5,.5
33)	\$ BLOCKS O-S NOT NEEDED
34)	\$ BLOCK R
35)	END

TITLE:  
THREE-BAR TRUSS

CONTROL PARAMETERS;	
CALCULATION CONTROL,	NCALC = 3
NUMBER OF GLOBAL DESIGN VARIABLES,	NDV = 2
INPUT PARAMETER FOR APPROX. OPT.,	NAPRX = 1
NUMBER OF PARAMETRIC VARIABLES,	NPV = 0
INPUT INFORMATION PRINT CODE,	IPNPUT = 0
DEBUG PRINT CODE,	IPDBG = 0

Figure 4 Continued: COPES Example, Optimization Using Approximation Techniques

# CALCULATION CONTROL, NCALC

VALUE	MEANING
1	SINGLE ANALYSIS
2	OPTIMIZATION
3	APPROXIMATE OPTIMIZATION
4	PARAMETRIC STUDY

## \* \* OPTIMIZATION INFORMATION

### OPTIMIZATION CONTROL PARAMETERS

ISTRAT	IOPT	IONED	IPRINT	NSCAL	IOVRDR	IOVRDI
0	5	7	0	0	0	0

GLOBAL VARIABLE NUMBER OF OBJECTIVE = 4  
 MULTIPLIER (NEGATIVE INDICATES MINIMIZATION) = -0.1000E+01

### DESIGN VARIABLE INFORMATION

NON-ZERO INITIAL VALUE WILL OVER-RIDE MODULE INPUT

D. V. NO.	LOWER BOUND	UPPER BOUND	INITIAL VALUE
1	0.10000E-01	0.11000E+16	0.0
2	0.10000E-01	0.11000E+16	0.0

### DESIGN VARIABLES

ID	D. V. NO.	GLOBAL VAR. NO.	MULTIPLYING FACTOR
1	1	1	0.10000E+01
2	2	2	0.10000E+01
3	1	3	0.10000E+01

### CONSTRAINT INFORMATION

THERE ARE 1 CONSTRAINT SETS

ID	GLOBAL VAR. 1	GLOBAL VAR. 2	TYPE ID	LOWER BOUND	NORM. FACTOR	UPPER BOUND	NORM. FACTOR
1	5	10	0	-0.1500E+05	0.1500E+05	0.2000E+05	0.2000E+05

TOTAL NUMBER OF CONSTRAINED PARAMETERS = 6

Figure 4 Continued: COPES Example, Optimization Using Approximation Techniques

\* \* APPROXIMATE ANALYSIS/OPTIMIZATION INFORMATION

PRINT CONTROL,	IPAPRX =	1
NUMBER OF INPUT X-VECTORS,	NXI =	4
NUMBER OF INPUT X-F PAIRS,	NXFI =	0
X-VECTOR FROM ANALIZ,	NXA =	0
NOMINAL DESIGN,	INOM =	0
READ UNIT FOR X-VECTORS,	ISCRX =	5
READ UNIT FOR X-F PAIRS,	ISCRXF =	5
MINIMUM APPROXIMATING CYCLES,	KMIN =	1
MAXIMUM APPROXIMATING CYCLES,	KMAX =	12
MAXIMUM DESIGNS USED IN FIT,	NPMAX =	10
NOMINAL DESIGN PARAMETER,	JNOM =	10
TAYLER SERIES I.D. CODE,	MAXTRM =	3

DELTA-X BOUNDS FOR APPROXIMATE OPTIMIZATION  
 0.2000E+00 0.2000E+00

MULTIPLIER ON DELX,	XFACT1 =	0.1500E+01
MULTIPLIER ON DELX,	XFACT2 =	0.2000E+01

GLOBAL LOCATIONS OF FUNCTIONS

4 5 6 7 8 9 10

X-VECTORS INPUT FROM UNIT 5

NUMBER	1	DESIGN	1
0.8000E+00		0.6000E+00	

NUMBER	2	DESIGN	2
0.4000E+00		0.1000E+01	

NUMBER	3	DESIGN	3
0.1000E+01		0.2000E+00	

NUMBER	4	DESIGN	4
0.5000E+00		0.5000E+00	

\* \* ESTIMATED DATA STORAGE REQUIREMENTS

REAL			INTEGER		
INPUT	EXECUTION	AVAILABLE	INPUT	EXECUTION	AVAILABLE
42	255	5000	32	74	1000

Figure 4 Continued: COPES Example, Optimization Using Approximation Techniques

APPROXIMATE OPTIMIZATION ITERATION HISTORY

APPROXIMATING FUNCTION 1 IS THE OBJECTIVE

BEGIN ITERATION NUMBER 1

NOMINAL DESIGN NUMBER = 1

X-VECTOR

0.80000E+00 0.60000E+00

FUNCTION VALUES

0.28627E+02 0.18566E+05 0.12132E+05 -0.64340E+04 -0.64340E+04  
0.12132E+05 0.18566E+05

CONDITIONING NUMBER FOR APPROXIMATION, COND = 0.23630E-01

RESULTS OF APPROXIMATE OPTIMIZATION

DELTA-X VECTOR

-0.52212E-01 -0.62160E-01

X-VECTOR

0.74779E+00 0.53784E+00

APPROXIMATE FUNCTION VALUES

0.26529E+02 0.20000E+05 0.13474E+05 -0.65257E+04 -0.65257E+04  
0.13474E+05 0.20000E+05

PRECISE FUNCTION VALUES

0.26529E+02 0.20002E+05 0.13259E+05 -0.67433E+04 -0.67433E+04  
0.13259E+05 0.20002E+05

Figure 4 Continued: COPES Example, Optimization Using Approximation  
Techniques

BEGIN ITERATION NUMBER 2

NOMINAL DESIGN NUMBER = 5

X-VECTOR  
0.74779E+00 0.53784E+00

FUNCTION VALUES  
0.26529E+02 0.20002E+05 0.13259E+05 -0.67433E+04 -0.67433E+04  
0.13259E+05 0.20002E+05

CONDITIONING NUMBER FOR APPROXIMATION, COND = 0.76112E-03

RESULTS OF APPROXIMATE OPTIMIZATION

DELTA-X VECTOR  
0.64294E-01 -0.30000E+00

X-VECTOR  
0.81208E+00 0.23784E+00

APPROXIMATE FUNCTION VALUES  
0.25348E+02 0.19996E+05 0.15840E+05 -0.41560E+04 -0.41560E+04  
0.15840E+05 0.19996E+05

PRECISE FUNCTION VALUES  
0.25348E+02 0.21021E+05 0.17415E+05 -0.36066E+04 -0.36066E+04  
0.17415E+05 0.21021E+05

Figure 4 Continued: COPES Example, Optimization Using Approximation  
Techniques



BEGIN ITERATION NUMBER 3

NOMINAL DESIGN NUMBER = 6

X-VECTOR

0.81208E+00 0.23784E+00

FUNCTION VALUES

0.25348E+02 0.21021E+05 0.17415E+05 -0.36066E+04 -0.36066E+04  
0.17415E+05 0.21021E+05

CONDITIONING NUMBER FOR APPROXIMATION, COND = 0.13815E-06

RESULTS OF APPROXIMATE OPTIMIZATION

DELTA-X VECTOR

-0.22536E-01 0.15137E+00

X-VECTOR

0.78955E+00 0.38921E+00

APPROXIMATE FUNCTION VALUES

0.26224E+02 0.20000E+05 0.14977E+05 -0.50230E+04 -0.50230E+04  
0.14977E+05 0.20000E+05

PRECISE FUNCTION VALUES

0.26224E+02 0.20128E+05 0.14926E+05 -0.52027E+04 -0.52027E+04  
0.14926E+05 0.20128E+05

Figure 4 Continued: COPES Example, Optimization Using Approximation  
Techniques

BEGIN ITERATION NUMBER 4

NOMINAL DESIGN NUMBER = 7

X-VECTOR

0.78955E+00 0.38921E+00

FUNCTION VALUES

0.26224E+02 0.20128E+05 0.14926E+05 -0.52027E+04 -0.52027E+04  
0.14926E+05 0.20128E+05

CONDITIONING NUMBER FOR APPROXIMATION, COND = 0.24905E-05

RESULTS OF APPROXIMATE OPTIMIZATION

OPTIMIZATION HAS PRODUCED AN X-VECTOR WHICH IS THE SAME AS A PREVIOUS DESIGN

DELTA-X VECTOR

0.0 0.0

X-VECTOR

0.78955E+00 0.38921E+00

THE FOLLOWING DESIGN IS NOT THE APPROXIMATE OPTIMUM

DELTA-X VECTOR

0.80000E-02 0.77921E-02

X-VECTOR

0.79755E+00 0.39701E+00

APPROXIMATE FUNCTION VALUES

0.26528E+02 0.19871E+05 0.14711E+05 -0.51594E+04 -0.51594E+04  
0.14711E+05 0.19871E+05

PRECISE FUNCTION VALUES

0.26528E+02 0.19897E+05 0.14717E+05 -0.51801E+04 -0.51801E+04  
0.14717E+05 0.19897E+05

Figure 4 Continued: COPES Example, Optimization Using Approximation Techniques

BEGIN ITERATION NUMBER 5

NOMINAL DESIGN NUMBER = 7

X-VECTOR

0.78955E+00 0.38921E+00

FUNCTION VALUES

0.26224E+02 0.20128E+05 0.14926E+05 -0.52027E+04 -0.52027E+04  
0.14926E+05 0.20128E+05

CONDITIONING NUMBER FOR APPROXIMATION, COND = 0.25034E-05

RESULTS OF APPROXIMATE OPTIMIZATION

DELTA-X VECTOR

0.47157E-02 0.16863E-02

X-VECTOR

0.79426E+00 0.39090E+00

APPROXIMATE FUNCTION VALUES

0.26374E+02 0.20000E+05 0.14845E+05 -0.51550E+04 -0.51550E+04  
0.14845E+05 0.20000E+05

PRECISE FUNCTION VALUES

0.26374E+02 0.20014E+05 0.14847E+05 -0.51668E+04 -0.51668E+04  
0.14847E+05 0.20014E+05

Figure 4 Continued: COPES Example, Optimization Using Approximation  
Techniques

BEGIN ITERATION NUMBER 6

NOMINAL DESIGN NUMBER = 9

X-VECTOR

0.79426E+00 0.39090E+00

FUNCTION VALUES

0.26374E+02 0.20014E+05 0.14847E+05 -0.51668E+04 -0.51668E+04  
0.14847E+05 0.20014E+05

CONDITIONING NUMBER FOR APPROXIMATION, COND = 0.23234E-05

RESULTS OF APPROXIMATE OPTIMIZATION

OPTIMIZATION HAS PRODUCED AN X-VECTOR WHICH IS THE SAME AS A PREVIOUS DESIGN

DELTA-X VECTOR

0.0 0.0

X-VECTOR

0.79426E+00 0.39090E+00

THE FOLLOWING DESIGN IS NOT THE APPROXIMATE OPTIMUM

DELTA-X VECTOR

0.80000E-02 0.78090E-02

X-VECTOR

0.80226E+00 0.39871E+00

APPROXIMATE FUNCTION VALUES

0.26678E+02 0.19758E+05 0.14634E+05 -0.51241E+04 -0.51241E+04  
0.14634E+05 0.19758E+05

PRECISE FUNCTION VALUES

0.26678E+02 0.19785E+05 0.14640E+05 -0.51448E+04 -0.51448E+04  
0.14640E+05 0.19785E+05

Figure 4 Continued: COPES Example, Optimization Using Approximation Techniques

BEGIN ITERATION NUMBER 7

NOMINAL DESIGN NUMBER = 9

X-VECTOR

0.79426E+00 0.39090E+00

FUNCTION VALUES

0.26374E+02 0.20014E+05 0.14847E+05 -0.51668E+04 -0.51668E+04  
0.14847E+05 0.20014E+05

CONDITIONING NUMBER FOR APPROXIMATION, COND = 0.23355E-05

SIDE CONSTRAINTS ON APPROXIMATE OPTIMIZATION

LOWER BOUNDS

-0.10000E+00 -0.10000E+00

UPPER BOUNDS

0.10000E+00 0.10000E+00

RESULTS OF APPROXIMATE OPTIMIZATION

DELTA-X VECTOR

0.0 0.0

X-VECTOR

0.79426E+00 0.39090E+00

APPROXIMATE FUNCTION VALUES

0.26374E+02 0.20014E+05 0.14847E+05 -0.51668E+04 -0.51668E+04  
0.14847E+05 0.20014E+05

TWO CONSECUTIVE APPROXIMATE OPTIMIZATIONS HAVE PRODUCED THE SAME DESIGN  
OPTIMIZATION TERMINATED

Figure 4 Continued: COPES Example, Optimization Using Approximation  
Techniques

FINAL RESULT OF APPROXIMATE OPTIMIZATION

NOMINAL DESIGN NUMBER = 9

X-VECTOR

0.79426E+00 0.39090E+00

FUNCTION VALUES

0.26374E+02 0.20014E+05 0.14847E+05 -0.51668E+04 -0.51668E+04  
0.14847E+05 0.20014E+05

CONDITIONING NUMBER FOR APPROXIMATION, COND = 0.23355E-05

RESULTS OF APPROXIMATE ANALYSIS/OPTIMIZATION

TITLE

THREE-BAR TRUSS

GLOBAL LOCATIONS OF INDEPENDENT X-VARIABLES

1 2

GLOBAL LOCATIONS OF FUNCTIONS, F(X)

4 5 6 7 8 9 10

APPROXIMATION IS BASED ON 10 DESIGNS

NOMINAL DESIGN IS DESIGN NUMBER 9

VALUES OF X-VARIABLES

0.7943E+00 0.3909E+00

VALUES OF FUNCTIONS, F(X)

0.2637E+02 0.2001E+05 0.1485E+05 -0.5167E+04 -0.5167E+04  
0.1485E+05 0.2001E+05

Figure 4 Continued: COPES Example, Optimization Using Approximation  
Techniques

COEFFICIENTS OF TAYLOR SERIES EXPANSION

PARAMETER 1 = GLOBAL VARIABLE 4

LINEAR TERMS, DEL F

0.2828E+02 0.1000E+02

NON-LINEAR TERMS, H, BEGINNING WITH DIAGONAL ELEMENT

ROW 1

0.1767E-02 0.9755E-03

ROW 2

0.3984E-03

PARAMETER 2 = GLOBAL VARIABLE 5

LINEAR TERMS, DEL F

-0.2412E+05 -0.8268E+04

NON-LINEAR TERMS, H, BEGINNING WITH DIAGONAL ELEMENT

ROW 1

0.4862E+05 0.4102E+04

ROW 2

0.1735E+05

PARAMETER 3 = GLOBAL VARIABLE 6

LINEAR TERMS, DEL F

-0.1129E+05 -0.1602E+05

NON-LINEAR TERMS, H, BEGINNING WITH DIAGONAL ELEMENT

ROW 1

0.1633E+05 0.2327E+05

ROW 2

0.3131E+05

Figure 4 Continued: COPES Example, Optimization Using Approximation Techniques

PARAMETER 4 = GLOBAL VARIABLE 7

LINEAR TERMS, DEL F  
0.1283E+05 -0.7753E+04

NON-LINEAR TERMS, H, BEGINNING WITH DIAGONAL ELEMENT

ROW 1  
-0.3229E+05 0.1917E+05

ROW 2  
0.1396E+05

PARAMETER 5 = GLOBAL VARIABLE 8

LINEAR TERMS, DEL F  
0.1283E+05 -0.7753E+04

NON-LINEAR TERMS, H, BEGINNING WITH DIAGONAL ELEMENT

ROW 1  
-0.3229E+05 0.1917E+05

ROW 2  
0.1396E+05

PARAMETER 6 = GLOBAL VARIABLE 9

LINEAR TERMS, DEL F  
-0.1129E+05 -0.1602E+05

NON-LINEAR TERMS, H, BEGINNING WITH DIAGONAL ELEMENT

ROW 1  
0.1633E+05 0.2327E+05

ROW 2  
0.3131E+05

Figure 4 Continued: COPES Example, Optimization Using Approximation Techniques



PARAMETER 7 = GLOBAL VARIABLE 10

LINEAR TERMS, DEL F  
-0.2412E+05 -0.8268E+04

NON-LINEAR TERMS, H, BEGINNING WITH DIAGONAL ELEMENT

ROW 1  
0.4862E+05 0.4102E+04

ROW 2  
0.1735E+05

#### OPTIMIZATION RESULTS

OBJECTIVE FUNCTION  
GLOBAL LOCATION 4 FUNCTION VALUE 0.26374E+02

#### DESIGN VARIABLES

ID	D. V. NO.	GLOBAL VAR. NO.	LOWER BOUND	VALUE	UPPER BOUND
1	1	1	-0.51448E+04	0.79426E+00	0.11000E+1
2	2	2	0.14640E+05	0.39090E+00	0.11000E+1
3	1	3	-0.51448E+04	0.79426E+00	0.11000E+1

#### DESIGN CONSTRAINTS

ID	GLOBAL VAR. NO.	LOWER BOUND	VALUE	UPPER BOUND
1	5	-0.15000E+05	0.20014E+05	0.20000E+05
3	6	-0.15000E+05	0.14847E+05	0.20000E+05
5	7	-0.15000E+05	-0.51668E+04	0.20000E+05
7	8	-0.15000E+05	-0.51668E+04	0.20000E+05
9	9	-0.15000E+05	0.14847E+05	0.20000E+05
11	10	-0.15000E+05	0.20014E+05	0.20000E+05

Figure 4 Continued: COPES Example, Optimization Using Approximation Techniques

### THREE-BAR TRUSS OUTPUT

A1 = 0.79426E+00  
A2 = 0.39090E+00  
A3 = 0.79426E+00

### STRESSES

SIG11 = 0.20014E+05  
SIG21 = 0.14847E+05  
SIG31 = -0.51668E+04  
SIG12 = -0.51668E+04  
SIG22 = 0.14847E+05  
SIG32 = 0.20014E+05

VOLUME = 0.26374E+02

### PROGRAM CALLS TO ANALIZ

ICALC	CALLS
1	1
2	10
3	1

Figure 4 Concluded: COPES Example, Optimization Using Approximation Techniques

CCCCCCC	0000000	PPPPPPP	EEEEEEE	SSSSSSS
C	0 0	P P	E	S
C	0 0	P P	E	S
C	0 0	PPPPPPP	EEEE	SSSSSSS
C	0 0	P	E	S
C	0 0	P	E	S
CCCCCCC	0000000	P	EEEEEEE	SSSSSSS

C O P E S / A D S

C O N T R O L   P R O G R A M

F O R

E N G I N E E R I N G   S Y N T H E S I S

U S I N G   T H E

A D S   O P T I M I Z A T I O N   P R O G R A M

T I T L E

THREE-BAR TRUSS

CARD IMAGES OF INPUT DATA

CARD	IMAGE
1)	THREE-BAR TRUSS
2)	\$ BLOCK B
3)	4,,3
4)	\$ BLOCKS C-O NOT NEEDED
5)	\$ BLOCK P
6)	7
7)	4,5,6,7,8,9,10
8)	\$ BLOCK Q
9)	1,3
10)	1.,.8,1.2
11)	2,3
12)	1.,.8,1.2
13)	3,5
14)	1.,.6,.8,1.2,1.4
15)	\$ BLOCK R
16)	END

Figure 5: COPES Example, Parametric Study

TITLE:  
THREE-BAR TRUSS

CONTROL PARAMETERS;

CALCULATION CONTROL,	NCALC =	4
NUMBER OF GLOBAL DESIGN VARIABLES,	NDV =	0
INPUT PARAMETER FOR APPROX. OPT.,	NAPRX =	0
NUMBER OF PARAMETRIC VARIABLES,	NPV =	3
INPUT INFORMATION PRINT CODE,	IPNPUT =	0
DEBUG PRINT CODE,	IPDBG =	0

CALCULATION CONTROL, NCALC

VALUE	MEANING
1	SINGLE ANALYSIS
2	OPTIMIZATION
3	APPROXIMATE OPTIMIZATION
4	PARAMETRIC STUDY

\* \* PARAMETRIC INFORMATION

PRINT CONTROL,	IPRNTP =	0
NUMBER OF FUNCTIONS,	NPF =	7

GLOBAL NUMBERS ASSOCIATED WITH PARAMETRIC FUNCTIONS

4	5	6	7	8	9	10
---	---	---	---	---	---	----

PARAMETERS

NUMBER	GLOBAL VARIABLE	NOMINAL VALUE	OFF-NOMINAL VALUES			
1	1	0.10000E+01	0.8000E+00	0.1200E+01		
2	2	0.10000E+01	0.8000E+00	0.1200E+01		
3	3	0.10000E+01	0.6000E+00	0.8000E+00	0.1200E+01	0.1400E+01

\* \* ESTIMATED DATA STORAGE REQUIREMENTS

REAL			INTEGER		
INPUT	EXECUTION	AVAILABLE	INPUT	EXECUTION	AVAILABLE
18	25	5000	14	14	1000

TITLE  
THREE-BAR TRUSS

NUMBER OF PARAMETRIC VARIABLES,	NPV =	3
NUMBER OF FUNCTIONS,	NPF =	7

GLOBAL LOCATIONS OF PARAMETRIC VARIABLES

1	2	3
---	---	---

GLOBAL LOCATIONS OF FUNCTIONS

4	5	6	7	8	9	10
---	---	---	---	---	---	----

Figure 5 Continued: COPES Example, Parametric Study

# NOMINAL DESIGN INFORMATION

## VALUES OF THE PARAMETERS

0.10000E+01 0.10000E+01 0.10000E+01

## VALUES OF THE FUNCTIONS

0.38284E+02 0.14142E+05 0.82843E+04 -0.58579E+04 -0.58579E+04  
0.82843E+04 0.14142E+05

## PARAMETRIC STUDY RESULTS

### GLOBAL VARIABLE 1

X	F(X)			
0.8000E+00	0.3546E+02 -0.6823E+04	0.1647E+05 0.7719E+04	0.9649E+04 0.1454E+05	-0.6823E+04
0.1200E+01	0.4111E+02 -0.5132E+04	0.1239E+05 0.8709E+04	0.7258E+04 0.1384E+05	-0.5132E+04

### GLOBAL VARIABLE 2

X	F(X)			
0.8000E+00	0.3628E+02 -0.5308E+04	0.1469E+05 0.9384E+04	0.9384E+04 0.1469E+05	-0.5308E+04
0.1200E+01	0.4028E+02 -0.6292E+04	0.1371E+05 0.7415E+04	0.7415E+04 0.1371E+05	-0.6292E+04

### GLOBAL VARIABLE 3

X	F(X)			
0.6000E+00	0.3263E+02 -0.8168E+04	0.1510E+05 0.1155E+05	0.6931E+04 0.1972E+05	-0.8168E+04
0.8000E+00	0.3546E+02 -0.6823E+04	0.1454E+05 0.9649E+04	0.7719E+04 0.1647E+05	-0.6823E+04
0.1200E+01	0.4111E+02 -0.5132E+04	0.1384E+05 0.7258E+04	0.8709E+04 0.1239E+05	-0.5132E+04
0.1400E+01	0.4394E+02 -0.4566E+04	0.1361E+05 0.6458E+04	0.9041E+04 0.1102E+05	-0.4566E+04

## PROGRAM CALLS TO ANALIZ

ICALC	CALLS
1	1
2	9
3	0

Figure 5 Concluded: COPES Example, Parametric Study

```

THREE-BAR TRUSS
$ BLOCK B
3,2,1,3
$ BLOCK C
0,5,7,0
$ BLOCK D
3,4,-1.
$ BLOCKS E AND F NOT NEEDED
$ BLOCK G
.01,1.0+20
.01,1.0+20
$ BLOCK H
1,1
2,2
1,3
$ BLOCK I
1
$ BLOCK J
5,10
-15000.,,20000.
$ BLOCK K
1,4
0
$ BLOCK L
.2,.2
$ BLOCK M
0.
$ BLOCK N
.8,.6
.4,1.
1.,.2
.5,.5
$ BLOCK O, NO DATA BECAUSE NXFI=0
$ BLOCK P
7
4,5,6,7,8,9,10
$ BLOCK Q
1,3
1.,.8,1.2
2,3
1.,.8,1.2
3,5
1.,.6,.8,1.2,1.4
$ BLOCK R
END

```

Figure 6: COPES Example, Combined Data For All Options

## 7.0 REFERENCES

1. Madsen, L. E. and Vanderplaats, G. N., "COPES - A FORTRAN Control Program for Engineering Synthesis," NPS69-81-003, Naval Postgraduate School, Monterey, Calif., March 1982.
2. Vanderplaats, G. N., "CONMIN - A FORTRAN Program for Constrained Function Minimization, User's Manual," NASA TM X-62,282, Aug. 1973.
3. Vanderplaats, G. N., Sugimoto, H. and Sprague, C. M., "ADS-1: A New General-Purpose Optimization Program," Proc. 24th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference, Lake Tahoe, Nev., May 1983.
4. Vanderplaats, G. N., "ADS - A FORTRAN Program for Automated Design Synthesis, Version 1.00," User's Manual, Department of Mechanical Engineering, Naval Postgraduate School, Monterey, Calif., May 1984.
5. Vanderplaats, G. N., "An Efficient Algorithm for Numerical Airfoil Optimization," AIAA J. Aircraft, Vol. 16, No. 12, Dec. 1979.
6. Garberoglio, J. E., Song, J. O. and Boudreaux, W. L., "Optimization of Compressor Vane and Bleed Settings," ASME Paper No. 82-GT-81, Proc. 27th International Gas Turbine Conference and Exhibit, London, April 18-22, 1982.
7. Schmit, L. A., "Structural Design by Systematic Synthesis," Proc. 2nd Conference on Electronic Computation, ASCE, New York, 1960.

## APPENDIX A

### ADS PROGRAM OPTIONS

The options available in the ADS program are identified here. This information is taken from the ADS User's manual (ref. 4).

#### A.1 INTRODUCTION

ADS is a general purpose numerical optimization program containing a wide variety of algorithms. The problem solved is:

Minimize  $F(\mathbf{X})$

Subject to;

$$g_j(\mathbf{X}) \leq 0 \quad j=1,m$$

$$h_k(\mathbf{X}) = 0 \quad k=1,l$$

$$x_i^l \leq x_i \leq x_i^u \quad i=1,n$$

The solution of this general problem is separated into three basic levels:

1. STRATEGY - For example, Sequential Unconstrained Minimization or Sequential Linear Programming.
2. OPTIMIZER - For example, Variable Metric methods for unconstrained minimization or the Method of Feasible Directions for constrained minimization.
3. ONE-DIMENSIONAL SEARCH - For example, Golden Section or Polynomial Interpolation.

Additionally, we may consider another component to be problem formulation. It is assumed that the engineer makes every effort to formulate the problem in a form amenable to efficient solution by numerical optimization. This aspect is perhaps the most important ingredient to the efficient use of the ADS program for solution of problems of practical significance.

By choosing the Strategy, Optimizer and One-Dimensional Search, the user is given considerable flexibility in creating an optimization program which works well for a given class of design problems.

The purpose here is to describe the available program options. Sections A.2-A4 identify the available optimization strategies, optimizers and one-dimensional search algorithms. Section A.5 gives allowed combinations of algorithms and a Quick-Reference summary. Section A.6 gives a brief set of guidelines for choosing an overall optimization algorithm.



## A.2 Strategy

Table A.1 lists the strategies available. The parameter ISTRAT will be sent to the ADS program to identify the strategy the user wants. The ISTRAT=0 option would indicate that control should transfer directly to the optimizer. This would be the case, for example, when using the Method of Feasible Directions to solve constrained optimization problems because the optimizer works directly with the constrained problem. On the other hand, if the constrained optimization problem is to be solved by creating a sequence of unconstrained minimizations, with penalty functions to deal with constraints, one of the appropriate strategies would be used.

TABLE A.1: STRATEGY OPTIONS

ISTRAT	STRATEGY TO BE USED
0	None. Go directly to the optimizer.
1	Sequential unconstrained minimization using the exterior penalty function method.
2	Sequential unconstrained minimization using the linear extended interior penalty function method.
3	Sequential unconstrained minimization using the quadratic extended interior penalty function method.
4	Sequential unconstrained minimization using the cubic extended interior penalty function method.
5	Augmented Lagrange Multiplier method.
6	Sequential Linear Programming.
7	Method of Centers (method of inscribed hyperspheres)
8	Sequential Quadratic Programming.

## A.3 Optimizer

Table A.2 lists the optimizers available. IOPT is the parameter used to indicate the optimizer desired.

TABLE A.2: OPTIMIZER OPTIONS

IOPT	OPTIMIZER TO BE USED
0	None. Go directly to the one-dimensional search. This option should be used only for program development.
1	Fletcher-Reeves algorithm for unconstrained minimization
2	Davidon-Fletcher-Powell (DFP) variable metric method for unconstrained minimization.
3	Broydon-Fletcher-Goldfarb-Shanno (BFGS) variable metric method for unconstrained minimization.
4	Method of Feasible Directions (MFD) for constrained minimization.
5	Modified Method of Feasible Directions for constrained minimization.

In choosing the optimizer (as well as strategy and one-dimensional search) it is assumed that the user is knowledgeable enough to choose an algorithm consistent with the problem at hand. For example, a variable metric optimizer would not be used to solve constrained problems unless a strategy is used to create the equivalent unconstrained minimization task via some form of penalty function.

#### A.4 One-Dimensional Search

Table A.3 lists the one-dimensional search options available for unconstrained and constrained problems. Here IONED identifies the algorithm to be used.

TABLE A.3: ONE-DIMENSIONAL SEARCH OPTIONS

IONED	ONE-DIMENSIONAL SEARCH OPTION
1	Find the minimum of an unconstrained function using the Golden Section method.
2	Find the minimum of an unconstrained function using the Golden Section method followed by polynomial interpolation.
3	Find the minimum of an unconstrained function by first finding bounds and then using polynomial interpolation.
4	Find the minimum of an unconstrained function by polynomial interpolation/extrapolation without first finding bounds on the solution.
5	Find the minimum of an constrained function using the Golden Section method.
6	Find the minimum of an constrained function using the Golden Section method followed by polynomial interpolation.
7	Find the minimum of an constrained function by first finding bounds and then using polynomial interpolation.
8	Find the minimum of an constrained function by polynomial interpolation/extrapolation without first finding bounds on the solution.

#### A.5 Allowable Combinations of Algorithms

Not all combinations of strategy, optimizer and one-dimensional search are meaningful. For example, constrained one-dimensional search is not meaningful when minimizing unconstrained functions.

Table A.4 identifies the combinations of algorithms which are available in the ADS program. In this table, an X is used to denote an acceptable combination of strategy, optimizer and one-dimensional search. An example is shown by the heavy line on the table which indicates that constrained optimization is to be performed by the Augmented Lagrange Multiplier Method (ISTRAT=5), using the BFGS optimizer (IOPT=3) and polynomial interpolation with bounds for the one-dimensional search (IONED=3). From the table, it is clear that a large number of possible combinations of algorithms are available.

TABLE A.4: PROGRAM OPTIONS

STRATEGY	OPTIMIZER				
	1	2	3	4	5
0	X	X	X	X	X
1	X	X	X	0	0
2	X	X	X	0	0
3	X	X	X	0	0
4	X	X	X	0	0
⑤	X	X	ⓧ	0	0
6	0	0	0	X	X
7	0	0	0	X	X
8	0	0	0	X	X

ONE-D SEARCH	1	2	3	4	5
1	X	X	X	0	0
2	X	X	X	0	0
3	X	X	ⓧ	0	0
4	X	X	X	0	0
5	0	0	0	X	X
6	0	0	0	X	X
7	0	0	0	X	X
8	0	0	0	X	X

Table A.5 contains an annotated version of Table A.4 for convenient reference once the user is familiar with ADS.

TABLE A.5: QUICK REFERENCE TO ADS OPTIONS

		IOPT	OPTIMIZER				
		1	Fletcher-Reeves				
		2	Davidon-Fletcher-Powell (DFP)				
		3	Broydon-Fletcher-Goldfarb-Shanno (BFGS)				
		4	Method of Feasible Directions				
		5	Modified Method of Feasible Directions				

STRATEGY	ISTRAT	IOPT 1	2	3	4	5
None	0	X	X	X	X	X
SUMT, Exterior	1	X	X	X	0	0
SUMT, Linear Extended Interior	2	X	X	X	0	0
SUMT, Quadratic Extended Interior	3	X	X	X	0	0
SUMT, Cubic Extended Interior	4	X	X	X	0	0
Augmented Lagrange Multiplier Meth.	5	X	X	X	0	0
Sequential Linear Programming	6	0	0	0	X	X
Method of Centers	7	0	0	0	X	X
Sequential Quadratic Programming	8	0	0	0	X	X

ONE-DIMENSIONAL SEARCH	IONED					
Golden Section Method	1	X	X	X	0	0
Golden Section + Polynomial	2	X	X	X	0	0
Polynomial Interpolation (bounded)	3	X	X	X	0	0
Polynomial Extrapolation	4	X	X	X	0	0
Golden Section Method	5	0	0	0	X	X
Golden Section + Polynomial	6	0	0	0	X	X
Polynomial Interpolation (bounded)	7	0	0	0	X	X
Polynomial Extrapolation	8	0	0	0	X	X

NOTE: An X denotes an allowed combination of algorithms.

## A.6 Choosing An Algorithm

One difficulty with a program such as ADS, which provides numerous options, is that of picking the best combination of algorithms to solve a given problem. While it is not possible to provide a concise set of rules, some general guidelines are offered here based on the author's experience. The user is strongly encouraged to try many different options in order to gain familiarity with ADS and to improve the probability that the best combination of algorithms is found for the particular class of problems being solved.

### UNCONSTRAINED FUNCTIONS (NCON=0, Side Constraints OK)

ISTRAT=0

Is computer storage very limited?

Yes - IOPT=1. Are function evaluations expensive?

Yes - Is the objective known to be approximately quadratic?

Yes - IONED=4

No - IONED=3

No - IONED=1 or 2

No - Is the analysis iterative?

Yes - IOPT=3. Are function evaluations expensive?

Yes - Is the objective known to be approximately quadratic?

Yes - IONED=4

No - IONED=3

No - IONED=1 or 2

No - IOPT=2 or 3. Are function evaluations expensive?

Yes - Is the objective known to be approximately quadratic?

Yes - IONED=4

No - IONED=3

No - IONED=1 or 2

### CONSTRAINED FUNCTIONS

Are relative minima known to exist?

Yes - ISTRAT=1, IOPT=3. Are function evaluations expensive?

Yes - IONED=3

No - IONED=1 or 2

No - Are the objective and/or constraints highly nonlinear?

Yes - Are function evaluations expensive?

Yes - ISTRAT=0, IOPT=4, IONED=7

No - ISTRAT=2, 3 or 5, IOPT=2 or 3, IONED=1 or 2

No - Is the design expected to be fully-constrained?  
(i.e. NDV active constraints at the optimum)

Yes - ISTRAT=6, IOPT=5, IONED=6

No - Is the analysis iterative?

Yes - ISTRAT=0, IOPT=4, IONED=7 or

ISTRAT=8, IOPT=5, IONED=7

No - ISTRAT=0, IOPT=5, IONED=7 or

ISTRAT=8, IOPT=5, IONED=7

## GENERAL APPLICATIONS

Often little is known about the nature of the problem being solved. Based on experience with a wide variety of problems, a very direct approach is given here for using ADS. The following table of parameters is offered as a sequence of algorithms. When using ADS the first few times, the user may prefer to run the cases given here, rather than using the decision approach given above. It is assumed here that a constrained optimization problem is being solved. If the problem is unconstrained, ISTRAT=0, IOPT=3 and IONED=2 or 3 is recommended.

ISTRAT	IOPT	IONED	IPRINT
8	5	7	2200
0	5	7	2020
0	4	7	2020
6	5	6	2200
5	3	3	2200
2	3	3	2200
1	3	3	2200

## APPENDIX B

### OVER-RIDING ADS DEFAULT PARAMETERS

Various internal parameters are defined on the first call to ADS which work well for the "average" optimization task. However, it is often desirable to change these in order to gain maximum utility of the program. Various real and integer scalar parameters are stored in arrays **WK** and **IWK** respectively. Those which the user may wish to change are listed in Tables B.1 through B.4, together with their default values and definitions. If the user wishes to change any of these, the appropriate component of **WK** or **IWK** is simply re-defined. For example, if the relative convergence criterion, **DELOBJ**, is to be changed to 0.002, this is done by setting location 12 in **WK** to 0.002, because **WK(12)** contains the value of **DELOBJ**.

TABLE B.1: REAL PARAMETERS STORED IN ARRAY WK

PARAMETER	LOCATION	DEFAULT	MODULES WHERE USED		
			ISTRAT	IOPT	IONED
ALAMDZ	1	0.0	5	-	-
BETAMC	2	0.0	7	-	-
CT <sup>1</sup>	3	-0.03	-	4,5	-
CTL	4	-0.005	-	4,5	-
CTLMIN	5	0.001	-	4,5	-
CTMIN	6	0.01	-	4,5	-
DABALP <sup>2</sup>	7	0.0001	-	ALL	-
DABOBJ	8	ABS(F0)/10000	ALL	-	-
DABOBM	9	ABS(F0)/1000	ALL	-	-
DABSTR	10	ABS(F0)/10000	ALL	-	-
DELALP <sup>3</sup>	11	0.005	-	-	1,2,5,6
DELOBJ	12	0.001	-	ALL	-
DELOBM	13	0.01	ALL	-	-
DELSTR	14	0.001	ALL	-	-
DLOBJ1	15	0.1	-	ALL	-
DLOBJ2	16	1000.0	-	ALL	-
DX1	17	0.1	-	ALL	-
DX2	18	0.2	-	ALL	-
EPSPEN	19	-0.05	2,3,4	-	-
EXTRAP	20	5.0	-	-	ALL
FDCH	21	0.01	-	ALL	-
FDCHM	22	0.001	-	ALL	-
GMULTZ	23	10.0	8	-	-
PSAIZ	24	0.95	8	-	-
RMULT	25	5.0	1,5	-	-
RMVLMZ	26	0.2	6,7,8	-	-
RP	27	10.0	1,5	-	-
RPMAX	28	1.0E+10	1,5	-	-
RPMULT	29	0.2	1,5	-	-
RPPMIN	30	1.0E-10	2,3,4	-	-
RPPRIM	31	100.0	2,3,4	-	-
SCFO	32	1.0	ALL	ALL	ALL
SCLMIN	33	0.001	ALL	ALL	ALL
STOL	34	0.001	-	4,5	-
THETAZ	35	0.1	-	4,5	-
XMULT	36	2.618034	-	-	1,2,3,5,6,7
ZRO	37	0.00001	ALL	ALL	ALL

1 If IOPT=4, CT=-0.1

2 If IONED=3 or 8, DABALP=0.001

3 If IONED=3 or 8, DELALP=0.05

NOTE: F0 is the magnitude of the initial objective function.



TABLE B.2: DEFINITIONS OF REAL PARAMETERS CONTAINED IN ARRAY WK

PARAMETER	DEFINITION
ALAMDZ	Initial estimate of the Lagrange Multipliers in the Augmented Lagrange Multiplier Method.
BETAMC	Additional steepest descent fraction in the method of centers. After moving to the center of the hypersphere, a steepest descent move is made equal to BETAMC times the radius of the hypersphere.
CT	Constraint tolerance in the Method of Feasible Directions or the Modified Method of Feasible Directions. A constraint is active if its numerical value is more positive than CT.
CTL	Same as CT, but for linear constraints.
CTLMIN	Same as CTMIN, but for linear constraints.
CTMIN	Minimum constraint tolerance for nonlinear constraints. If a constraint is more positive than CTMIN, it is considered to be violated.
DABALP	Absolute convergence criteria for the one-dimensional search when using the Golden Section method.
DABOBJ	Maximum absolute change in the objective between two consecutive iterations to indicate convergence in optimization.
DABOBM	Absolute convergence criterion for the optimization sub-problem when using sequential minimization techniques.
DABSTR	Same as DABOBJ, but used at the strategy level.
DELALP	Relative convergence criteria for the one-dimensional search when using the Golden Section method.
DELOBJ	Maximum relative change in the objective between two consecutive iterations to indicate convergence in optimization.
DELOBM	Relative convergence criterion for the optimization sub-problem when using sequential minimization techniques.
DELSTR	Same as DELOBJ, but used at the strategy level.
DLOBJ1	Relative change in the objective function attempted on the first optimization iteration. Used to estimate initial move in the one-dimensional search. Updated as the optimization progresses.
DLOBJ2	Absolute change in the objective function attempted on the first optimization iteration. Used to estimate initial move in the one-dimensional search. Updated as the optimization progresses.
DX1	Maximum relative change in a design variable attempted on the first optimization iteration. Used to estimate the initial move in the one-dimensional search. Updated as the optimization progresses.
DX2	Maximum absolute change in a design variable attempted on the first optimization iteration. Used to estimate the initial move in the one-dimensional search. Updated as the optimization progresses.
EPSPEN	Initial transition point for extended penalty function methods. Updated as the optimization progresses.
EXTRAP	Maximum multiplier on the one-dimensional search parameter, ALPHA in the one-dimensional search using polynomial interpolation/extrapolation.

TABLE B.2 CONCLUDED: DEFINITIONS OF REAL PARAMETERS CONTAINED IN ARRAY WK

PARAMETER	DEFINITION
FDCH	Relative finite difference step when calculating gradients.
FDCHM	Minimum absolute value of the finite difference step when calculating gradients. This prevents too small a step when X(I) is near zero.
GMULTZ	Initial penalty parameter in Sequential Quadratic programming.
PSAIZ	Move fraction to avoid constraint violations in Sequential Quadratic Programming.
RMULT	Penalty function multiplier for the exterior penalty function method. Must be greater than 1.0.
RMVLMZ	Initial relative move limit. Used to set the move limits in sequential linear programming, method of inscribed hyperspheres and sequential quadratic programming as a fraction of the value of X(I), I=1,NDV.
RP	Initial penalty parameter for the exterior penalty function method or the Augmented Lagrange Multiplier method.
RPMAX	Maximum value of RP for the exterior penalty function method or the Augmented Lagrange Multiplier method.
RPMULT	Multiplier on RP for consecutive iterations.
RRPMIN	Minimum value of RPPRIM to indicate convergence.
RPPRIM	Initial penalty parameter for extended interior penalty function methods.
SCFO	The user-supplied value of the scale factor for the objective function if the default or calculated value is to be overridden.
SCLMIN	Minimum numerical value of any scale factor allowed.
STOL	Tolerance on the components of the calculated search direction to indicate that the Kuhn-Tucker conditions are satisfied.
THETAZ	Nominal value of the push-off factor in the Method of Feasible Directions.
XMULT	Multiplier on the move parameter, ALPHA, in the one-dimensional search to find bounds on the solution.
ZRO	Numerical estimate of zero on the computer. Usually the default value is adequate. If a computer with a short word length is used, ZRO=1.0E-4 may be preferred.

TABLE B.3: INTEGER PARAMETERS STORED IN ARRAY IWK

PARAMETER	LOCATION	DEFAULT	MODULES WHERE USED		
			ISTRAT	IOPT	IONED
ICNDIR	1	NDV+1	-	ALL	-
ISCAL	2	1	ALL	ALL	ALL
ITMAX	3	40	-	ALL	-
ITRMOP	4	3	-	1,2,3	-
ITRMST	5	2	ALL	-	-
JONED	6	IONED	8	-	-
JTMAX	7	20	ALL	-	-

TABLE B.4: DEFINITIONS OF INTEGER PARAMETERS CONTAINED IN ARRAY IWK

PARAMETER	DEFINITION
ICNDIR	Restart parameter for conjugate direction and variable metric methods. Unconstrained minimization is restarted with a steepest descent direction every ICNDIR iterations.
ISCAL	Scaling parameter. If ISCAL=0, no scaling is done. If ISCAL=1, the design variables, objective and constraints are scaled automatically.
ITMAX	Maximum number of iterations allowed at the optimizer level.
ITRMOP	The number of consecutive iterations for which the absolute or relative convergence criteria must be met to indicate convergence at the optimizer level.
ITRMST	The number of consecutive iterations for which the absolute or relative convergence criteria must be met to indicate convergence at the strategy level.
JONED	The one-dimensional search parameter (IONED) to be used in the Sequential Quadratic Programming method at the strategy level.
JTMAX	Maximum number of iterations allowed at the strategy level.

# INITIAL DISTRIBUTION LIST

	Number of Copies
1. Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, CA 93943	2
3. Office of Research Administration, Code 012 Naval Postgraduate School Monterey, CA 93943	1
4. Professor G. N. Vanderplaats Dept. Mechanical & Environmental Engineering University of California Santa Barbara, CA 93106	5
5. Dr. Young Shen Code 1542 David Taylor Naval Ship R&D Center Bethesda, MD 20084	10
6. Dr. Frank Peterson NAVSEA Systems Command Code 56XP Washington, DC 20362	10
7. Professor Robert H. Nunn, Code 69Nn Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943	10

**END**

**FILMED**

**1-85**

**DTIC**